Title: Correlation Noise Estimation in Distributed Video Coding

Authors Jürgen Slowack, Jozef Škorupa, Stefaan Mys, Nikos Deligiannis, Peter Lambert, Adrian Munteanu, and Rik Van de Walle

In: Effective Video Coding for Multimedia Applications, ISBN 978-953-307-177-0, pages 133-156, Intech Publishing, 2011.

0

Correlation Noise Estimation in Distributed Video Coding

Jürgen Slowack[1], Jozef Škorupa[1], Stefaan Mys[1], Nikos Deligiannis[2], Peter Lambert[1], Adrian Munteanu[2] and Rik Van de Walle[1]
[1]*Ghent University – IBBT*
Department of Electronics and Information Systems (ELIS) – Multimedia Lab
Gaston Crommenlaan 8 bus 201, B-9000 Ghent
[2]*Vrije Universiteit Brussel (VUB) – IBBT*
Electronics and Informatics Department (ETRO)
Pleinlaan 2, B-1050 Brussels
[1,2]*Belgium*

1. Introduction

Video compression is achieved by exploiting spatial and temporal redundancies in the frame sequence. In typical systems, the encoder is made responsible for exploiting the redundancies by predicting the current frame to be coded from previously coded information (such as other frames and/or blocks). Next, the residual between the frame to be coded and its prediction is transformed, quantized, and entropy coded. As the quality of the prediction has a large influence on the coding performance, high performing but computationally expensive algorithms for generating the prediction have been developed. As a result, typical video coding architectures show an imbalance, with an encoder that is significantly more complex than the decoder.

A new way for performing video coding has been introduced in the last decade. This new paradigm, called distributed video coding (DVC), shifts the complexity from the encoder to the decoder. Such a setup facilitates a different range of applications where the main focus and constraints are on the video (capturing and) coding devices, instead of on the decoding (and displaying) devices. Some examples of target applications include video conferencing with mobile devices, wireless sensor networks and multi-view video entertainment.

The aforementioned shift in complexity is realized by making the decoder responsible for generating the prediction, hereby relieving the encoder from this complex task. While the encoder has the ability to select the best prediction based on a comparison with the original to be coded, the decoder can not perform this comparison as it has only access to already decoded information, and not to the original. This complicates the decoder's task to estimate an accurate motion field compared to conventional predictive video coding.

In distributed video coding, the prediction generated at the decoder (called the side information) often contains a significant amount of errors in comparison to the original video frames. Therefore, the errors are corrected using error correcting information sent by the encoder (such as turbo or LDPC codes). For efficient use of these error correcting bits, soft channel information is needed at the decoder concerning the quality of the generated side

information Y with respect to the original X present at the encoder. More specifically, the decoder needs information expressing the correlation between X and Y. This correlation needs to be estimated since X is available only at the encoder, while Y is available only at the decoder.
The accuracy of estimating the correlation between X and Y has a large impact on compression performance in distributed video coding. When using a less accurate model, more rate from the encoder will be needed in order to correct the errors in Y and reliably decode X. Hence, one way to improve DVC compression performance is by focusing on the correlation model and by improving the accuracy of estimating it in practical DVC systems.
The correlation between X and Y is usually expressed through the difference $N = X - Y$, referred to as the correlation noise. Modeling the distribution of N is difficult because of its non-stationary characteristics, both in the temporal and spatial direction. This is illustrated by means of an example in Figure 1. In this example, the decoded frames at index 39 and 41 are used by the decoder to generate an estimation of the frame at index 40, following the techniques described in the context of the well-known DVC system called DISCOVER (Artigas et al. (2007)). When analyzing the correlation noise N in Figure 1, one can observe that N is spatially non-stationary, meaning that different spatial regions feature different error distributions. Some regions are well predicted (such as the grass in the background), while the prediction accuracy for other regions is rather poor. Similar non-stationary behavior can be observed in the temporal direction as well. Other sequences lead to similar conclusions, as illustrated by Figure 2 for the Table Tennis sequence. Due to these highly varying statistics, accurately estimating the distribution of N has proved to be a challenging task in distributed video coding.
In this chapter we describe several techniques that improve upon current approaches for correlation estimation in DVC. First, in Section 2, details are provided for the DVC architecture based on which our designs are built. This description is followed by a discussion on the current literature concerning correlation noise estimation, in Section 3. Next, two techniques that improve upon existing approaches are presented. The first technique (described in Section 4) incorporates knowledge about the quantization noise, which improves the accuracy of the correlation model, particularly at low rates. In the second improvement, we compensate for inaccurate assumptions made when generating the side information, as discussed in Section 5. The results achieved by both approaches are promising, and they underline the importance of accurate correlation modeling in DVC. Final conclusions of our work are given in Section 6.

2. Introducing the DVC architecture

Figure 3 depicts the DVC codec that is used as a starting point for the techniques presented in this chapter. The architecture is largely based on the pioneering architecture developed by Aaron et al. (2004a), and on its popular extension developed in the context of DISCOVER by Artigas et al. (2007). The latter can still be considered among the current state-of-the-art in DVC, and it provides a benchmark as its executables are available online[1].
The operation of the codec in Figure 3 is as follows. First, the frame sequence is partitioned into key frames I and Wyner-Ziv frames W, using a fixed GOP structure for the entire sequence (e.g., a GOP of length four would be: I-W-W-W-I-...). The key frames are coded without using other frames as references, i.e., applying H.264/AVC intra coding. Each

[1] http://www.discoverdvc.org

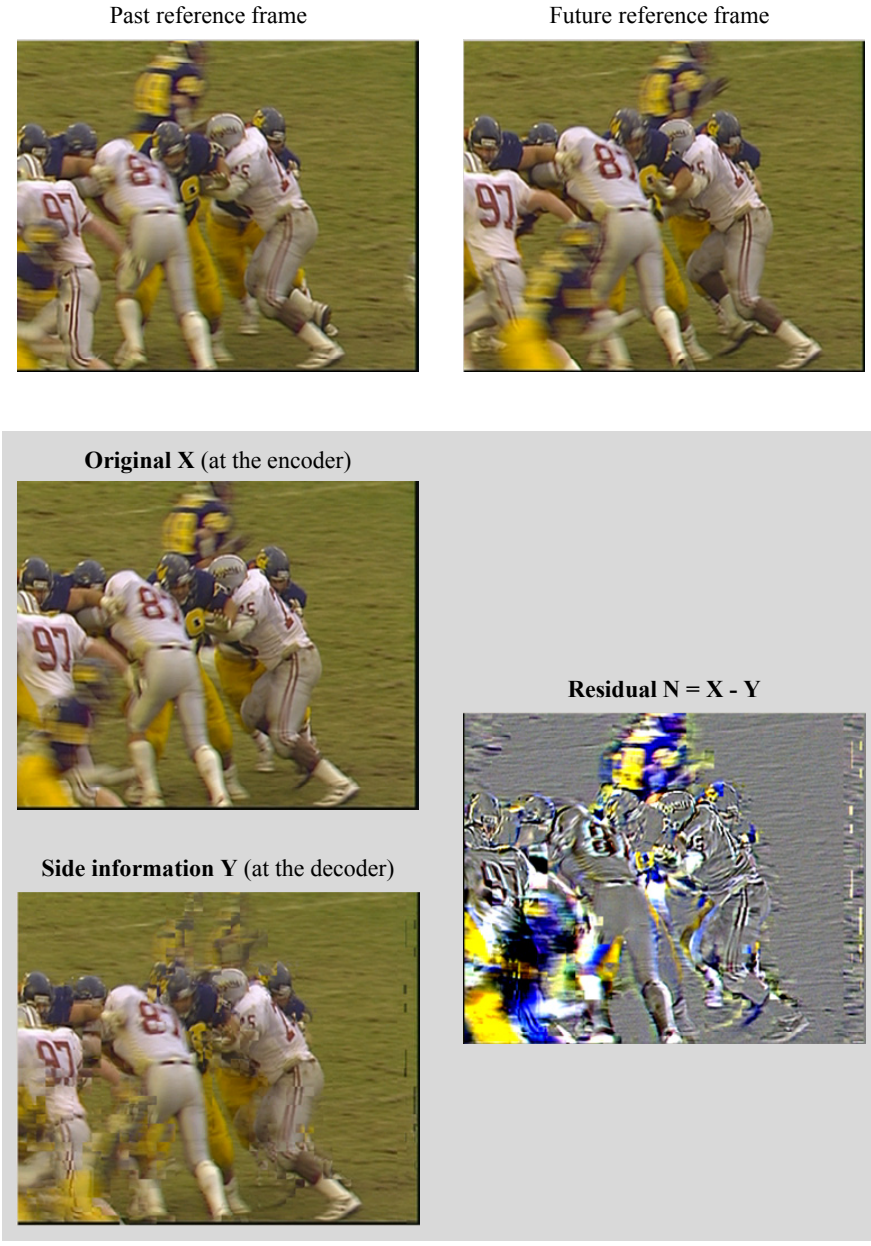

Fig. 1. Illustration of the correlation noise for the Football sequence, frame index 40 (past reference frame at index 39, future reference frame index 41).

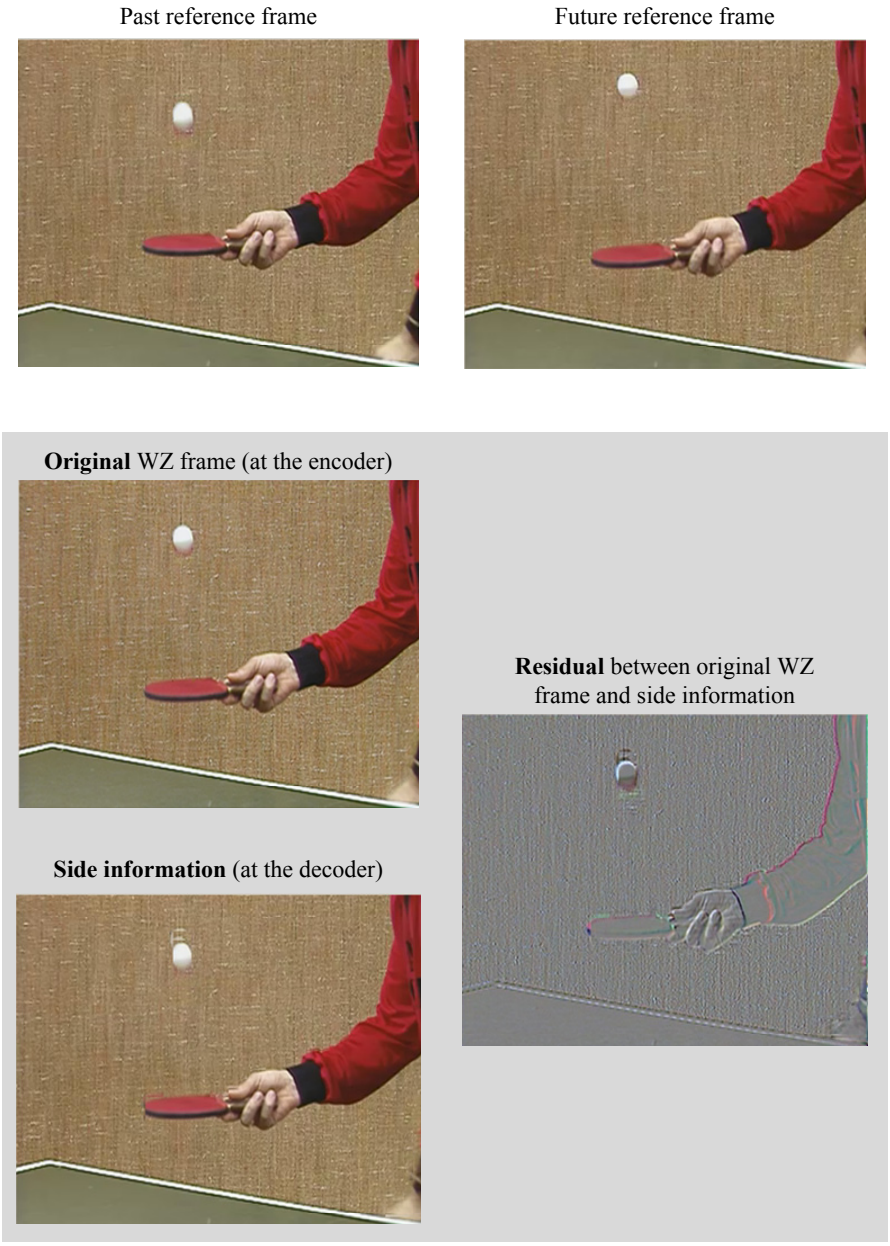

Fig. 2. Illustration of the correlation noise for the Table Tennis sequence, frame index 2 (past reference frame at index 1, future reference frame index 3).

Wyner-Ziv frame is partitioned into non-overlapping blocks of 4-by-4 pixels, and each block is transformed using a discrete cosine transform (DCT). Next, for all blocks, the resulting DCT coefficients at the same spatial index k (with $0 \leq k \leq 15$) are grouped into coefficient bands X_k. For example, each fourth coefficient in each block will be collected, forming the fourth coefficient band X_3. Next, each band is quantized to Q_k using a quantizer with 2^{M_k} quantization bins. The zero bin of this quantizer is 1.5 times larger than the other bins, for all coefficient bands except for the first (DC band). Subsequently, for each band, bits at identical positions are grouped into bitplanes BP_i^k. For example, all most significant bits of all DC coefficients will form bitplane BP_0^0. Finally, for each bitplane, parity bits are generated by a turbo coder (Lin & Costello (2004)) and stored in a buffer. These bits will be sent in portions to the decoder upon request.

At the decoder, key frames are decoded into I' by applying H.264/AVC intra decoding. We note that, in our notation, $'$ is used to indicate decoded frames. Side information is generated for each Wyner-Ziv frame, using already decoded frames as references. A hierarchical GOP structure is used, meaning that the sequence $I_1 - W_1 - W_2 - W_3 - I_2$ is coded and decoded in the following order: $I_1 - I_2 - W_2 - W_1 - W_3$. For example, the side information for W_1 will be generated using I_1' as a past reference frame, and W_2' as a future reference frame (Aaron et al. (2003)). Several techniques for generating the side information have been proposed in the literature. In our architecture, we follow the method adopted in DISCOVER (Artigas et al. (2007)).

After generating the side information Y, the correlation between X and Y is modeled. This correlation model – forming the main subject in this chapter – will be described in detail in Section 3.1, and it will be further extended in the remainder of this chapter.

The turbo decoder uses the correlation model in a Viterbi-like decoding procedure. To this extent, the turbo decoder requests parity bits (also called Wyner-Ziv bits) from the encoder's buffer via the feedback channel, until reliable decoding is achieved (Škorupa et al. (2009)).

When turbo decoding terminates, the bitplanes are multiplexed. This operation is the inverse of the bitplane extraction process performed at the encoder. As a result, the turbo decoder returns for each coefficient the index of the quantization bin containing the original value with very high probability. The following step – called reconstruction – is to select one particular value in this bin as the decoded value of the coefficient. The reconstruction method used here is the so-called centroid reconstruction, as described by Kubasov et al. (2007). After reconstruction, the result is inverse transformed, yielding the decoded Wyner-Ziv frame W'.

3. Related work on correlation estimation

The correlation between X and Y is commonly modeled using a Laplace distribution (Aaron et al. (2004a); Brites & Pereira (2008); Kubasov et al. (2007)), or a Gaussian distribution (Macchiavello et al. (2009)). The Laplace distribution is used by the majority of researchers as a good trade-off between model accuracy and complexity.

Using the Laplace distribution, the correlation between X and Y is described through a conditional probability density function of the form:

$$f_{X|Y}(x|y) = \frac{\alpha}{2} e^{-\alpha|x-y|}. \tag{1}$$

At the decoder, the realization of the side information, y, is available, and so only the distribution scale parameter α needs to be estimated. The relation between α and the variance σ^2 is given by:

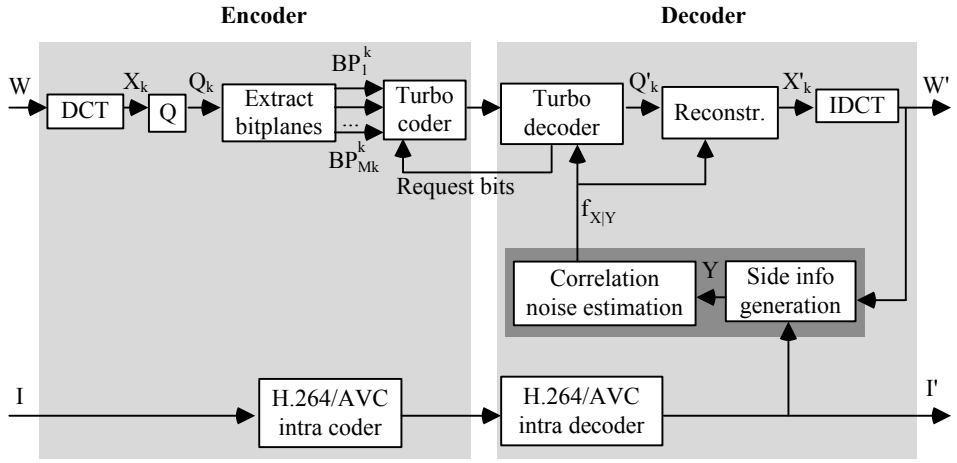

Fig. 3. The DVC architecture used as a starting point for the designs presented in this chapter.

$$\alpha = \frac{\sqrt{2}}{\sigma}. \tag{2}$$

Initially, stationary models and/or offline approaches have been used in the literature to estimate α. For example, in Stanford's DVC system, parameters for each coefficient band are obtained through an initial training stage performed on several sequences (Aaron et al. (2004a;b); Girod et al. (2005)). A temporally and spatially stationary model is adopted as the same parameter set is used throughout the entire sequence. Other offline techniques allow access to X at the decoder for obtaining correlation information (Trapanese et al. (2005)).

While these techniques are impractical or lack flexibility, efficient online techniques have been proposed only recently. An important contribution in this context is due to Brites & Pereira (2008), who propose offline and online methods both in the pixel-domain and transform-domain. Since each block in the side information is generated as the average between a past reference block and a future reference block, the similarity between these two reference blocks is used to estimate the correlation between the original X and its estimation Y. If the side information generation process is unable to find good matches between past and future reference blocks, then X and Y are assumed to be poorly correlated. On the other hand, if there is a good match, the correlation between X and Y is assumed to be strong. As such, this technique allows online estimation of the correlation noise by analyzing similarities between past and future motion-compensated frames.

An alternative solution has been proposed in our own work (Škorupa et al. (2008)). This technique – described further on in detail – also uses the motion-compensated residual between the past and future reference frames for estimating the correlation noise. However, one of the major differences with the previous technique is that the transform-domain noise is estimated by converting the pixel-domain noise estimates. The results show increased performance compared to the work of Brites & Pereira (2008).

Converting pixel-domain correlation noise estimates to their transform-domain equivalents has been proposed as well by Fan, Au & Cheung (2009). In the latter, instead of using the

motion-compensated residual, the authors exploit information available from the previously decoded Wyner-Ziv frame as well as the previously decoded coefficient bands.
Information from decoded coefficient bands is used also by Huang & Forchhammer (2009), aiming to improve the method proposed by Brites & Pereira (2008). The decoded information is used to classify coefficients, applying different estimators for different categories.
As discussed further, the techniques for correlation noise estimation described in the literature have a few shortcomings, especially the ones that exclusively rely on the motion-compensated residual. Therefore, in this chapter, two techniques are presented that improve upon previous approaches in the literature, including our previous work on correlation noise estimation (Škorupa et al. (2008)). To this extent, we first describe in detail this method in the following subsection.

3.1 From pixel-domain to transform-domain correlation estimation (Škorupa et al.)

Using common techniques for side information generation such as the ones used in DISCOVER (Artigas et al. (2007)), each block in the side information frame Y is created through motion compensation. More precisely, each block in Y is created as the average of a block in a decoded frame X'_B in the past (i.e., a frame at a lower frame-index relative to the current frame) and a block in a decoded frame X'_F in the future (i.e., a frame at a higher frame-index). The reference blocks are obtained by following the calculated past and future motion vectors denoted by (dx_B,dy_B) and (dx_F,dy_F), respectively.
As in the work of Brites & Pereira (2008), the similarity between past and future blocks is used to estimate the correlation noise. If the difference between the reference blocks is low, the corresponding block in the side information is assumed to be of high quality. On the other hand, if the difference between both blocks is large then it is likely that side information generation failed to obtain a reliable estimate.
Let R denote the motion-compensated residual given by:

$$R(x,y) = X'_B(x+dx_B,y+dy_B) - X'_F(x+dx_F,y+dy_F). \tag{3}$$

As such, there exists a relation between the correlation noise N and the motion-compensated residual R. This relation is illustrated by means of an example in Figure 4, which contains the result from coding frame 93 of the Foreman sequence (I-W-I-W GOP structure). This example shows that there is strong resemblance between R and N. Good matches between past and future reference blocks (i.e., low values for R) indeed often correspond to low values for N. Hence, although N can not be determined at the decoder in a practical DVC system, due to the absence of X, R can be calculated, since it only involves decoded reference frames. Therefore, similar to the work of Brites & Pereira (2008), we use R as basis for estimating N. However, the actual estimate of N differs in our method, as we first generate a pixel-domain estimation and then convert this result to the transform domain.
Using R, for each block at index k in the side information, the central moment is calculated:

$$Mom_k(R) = E_k\left[|R(x,y)|^{0.5}\right], \tag{4}$$

where E_k denotes the expectation taken only over the block at index k in the frame. Likewise, the average central moment $Mom(R)$ is obtained through expectation over the entire residual frame R:

$$Mom(R) = E\left[|R(x,y)|^{0.5}\right]. \tag{5}$$

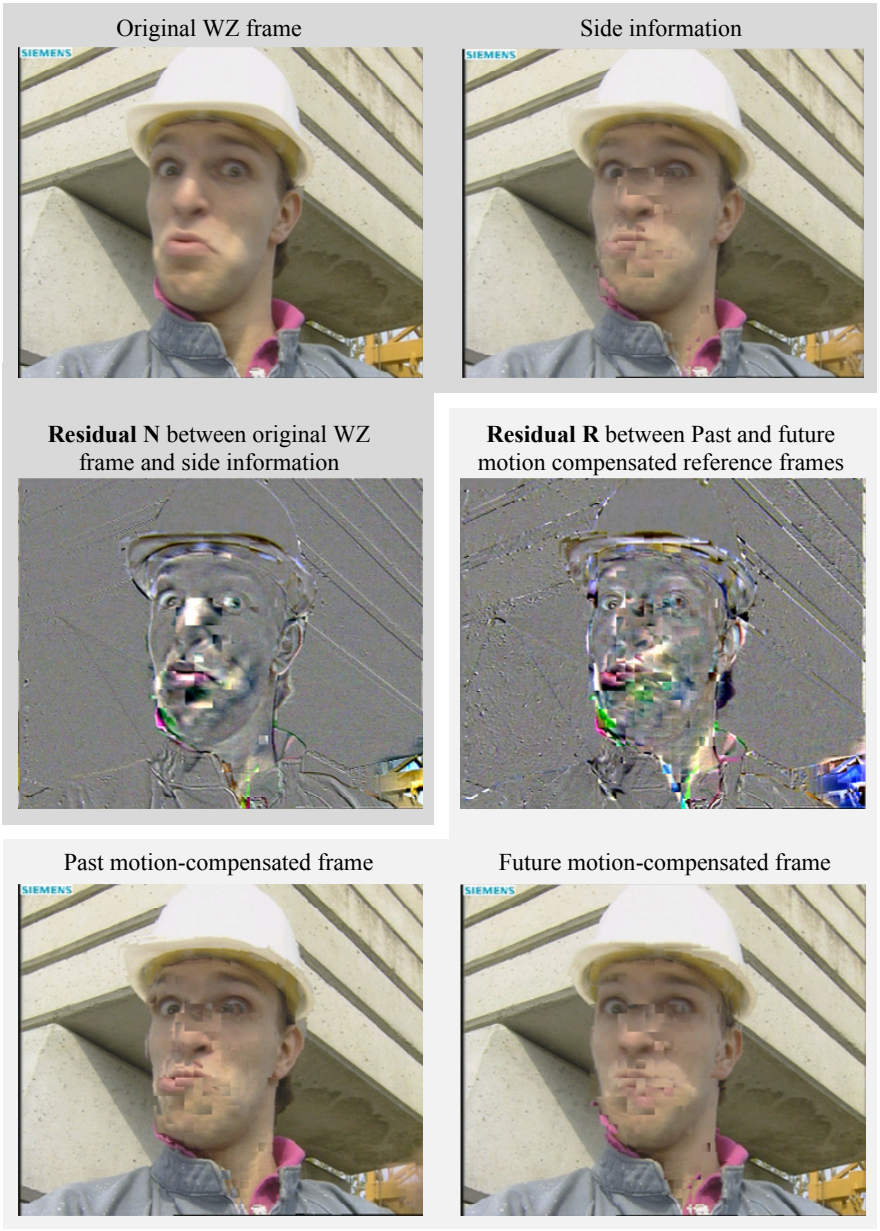

Fig. 4. Correlation noise N compared to the motion compensated residual R (fine quantization, intra quantization parameter of 10 is employed for the reference frames).

$s_{i,j}$	j			
i	4.25	2.06	1.16	0.77
	2.06	1.00	0.56	0.38
	1.16	0.56	0.32	0.21
	0.77	0.38	0.21	0.14

Table 1. Scaling parameters $s_{i,j}$ for pixel to transform-domain conversion of the α parameter estimation, following the techniques proposed in our previous work (Škorupa et al. (2008)).

The central moment of the correlation noise N for the block at index k is then estimated as:

$$\widetilde{Mom}_k(N) = \frac{Mom_k(R) + Mom(R)}{2}. \tag{6}$$

The rationale behind this formula is that it estimates the central moment of N for the block at index k by combining local and global information from R.
Finally, for each pixel in the block at index k, the following expression is used to estimate α:

$$\alpha_k^P = \frac{\pi}{4\widetilde{Mom}_k(N)^2}, \tag{7}$$

where the upper-index P indicates that this α parameter is defined in the pixel-domain. The lower index k differentiates between different blocks in the same frame, so as to cope with the spatial non-stationarities discussed before.
To convert the pixel-domain α parameter to its (DCT) transform-domain equivalent, a scaling step is applied. As such, the α parameter of the coefficient at index (i,j) in block k is given by:

$$\alpha_{k,(i,j)}^T = \frac{\alpha_k^P}{\sqrt{s_{i,j}}}, \tag{8}$$

with $s_{i,j}$ defined as in Table 1 ($0 \leq i \leq 3$ and $0 \leq j \leq 3$). For more information about this scaling operation we refer to Škorupa et al. (2008).
For future reference in the extensions provided in this chapter, we define the average pixel-domain α as:

$$\alpha^P = \frac{\pi}{4\widetilde{Mom}(N)^2}, \tag{9}$$

where $\widetilde{Mom}(N)$ denotes the frame-average of $\widetilde{Mom}_k(N)$. We also define its transform-domain counterpart as:

$$\alpha_{(i,j)}^T = \frac{\alpha^P}{\sqrt{s_{i,j}}}. \tag{10}$$

4. Accounting for quantization noise

As a first extension in this chapter, the model for correlation noise estimation is refined by accounting for the quantization noise. The technique is based on the observation that the quantization noise in the decoded reference frames X'_B and X'_F has a different impact on R than it has on N. As a result, inaccuracies occur in the previous method for correlation noise estimation when the quantization noise is high (i.e., for coarse quantization).

For fine quantization (Figure 4) of the intra-frames, the residual R and the correlation noise N show strong resemblance. In well-predicted areas both N and R depict low (i.e. mid-gray) values. On the other hand, R still provides reasonably accurate information about the average mismatch in areas that are poorly predicted. Hence, for fine quantization of the intra-frames, R can indeed be used to estimate N.
However, as shown in Figure 5, when the intra frames are coarsely quantized, there is a mismatch between R and N. In specific, the distribution of N has a much larger variance than the distribution of R. This is a consequence of the quantization noise present in the reference frames. Due to this noise, some of the fine detail is lost in the past and future reference frames. However, the side information generation process is still able to find good matches between past and future blocks, since the details have been erased in a similar way in both frames. As a result, the residual R is typically low, and the side information Y that is constructed through interpolation does not contain some of the fine details either. Consequently, the lost details can be found in N, but not in R, resulting in higher energy for N compared to the energy in R. For example, one can observe in Figure 5 that some of the texture details of the concrete in the background are present in N but not in R.
Our current technique (proposed in Škorupa et al. (2008) and described in Section 3.1) compensates insufficiently for quantization noise. To illustrate this, measurements have been performed for the luma DC component of Foreman (CIF, first 101 frames, I-W-I-W GOP structure). The distribution of N measured offline has been compared against the average noise distribution estimated online using the method described in Section 3.1. The results are presented in Figure 6, including the measured distribution of R. These results show that – for coarse quantization (i.e., IQP 40) – there is a clear mismatch between the measured distribution of N and its estimated distribution based on R.

4.1 Proposed solution

As a solution to this problem, statistics about the quantization noise are determined at the encoder. This information is then sent to the decoder, where it is used to improve the estimation of N.
For high resolution quantization, i.e., when the width of the quantization bin is small compared to the variance of the distribution of the signal, the average distortion due to uniform quantization can be approximated by the distortion of a random variable that is uniformly distributed over the quantization bin, which has a variance of $d^2/12$, where d denotes the bin width (Gersho & Gray (1992)). In our case, this approximation is inaccurate since medium and low rates are specifically targeted. At these rates, the quantization noise depends on the distribution of the signal, hence it is sequence-dependent, and non-stationary (in time and space[2]). Therefore, the quantization noise of the intra frames is calculated at the encoder (Section 4.1.1) and this information is used at the decoder to improve the estimation of the correlation noise (Section 4.1.2).

4.1.1 Encoder-side

For each coefficient band, the variance of the quantization noise of the intra frames is estimated by calculating the variance of the difference between the transformed original intra frame and the transformed intra decoded version. The computational overhead remains low,

[2]In the technique proposed here, the average quantization noise will be calculated per frame, hereby ignoring spatial differences. Accounting for spatial differences comes at the cost of increased signaling overhead.

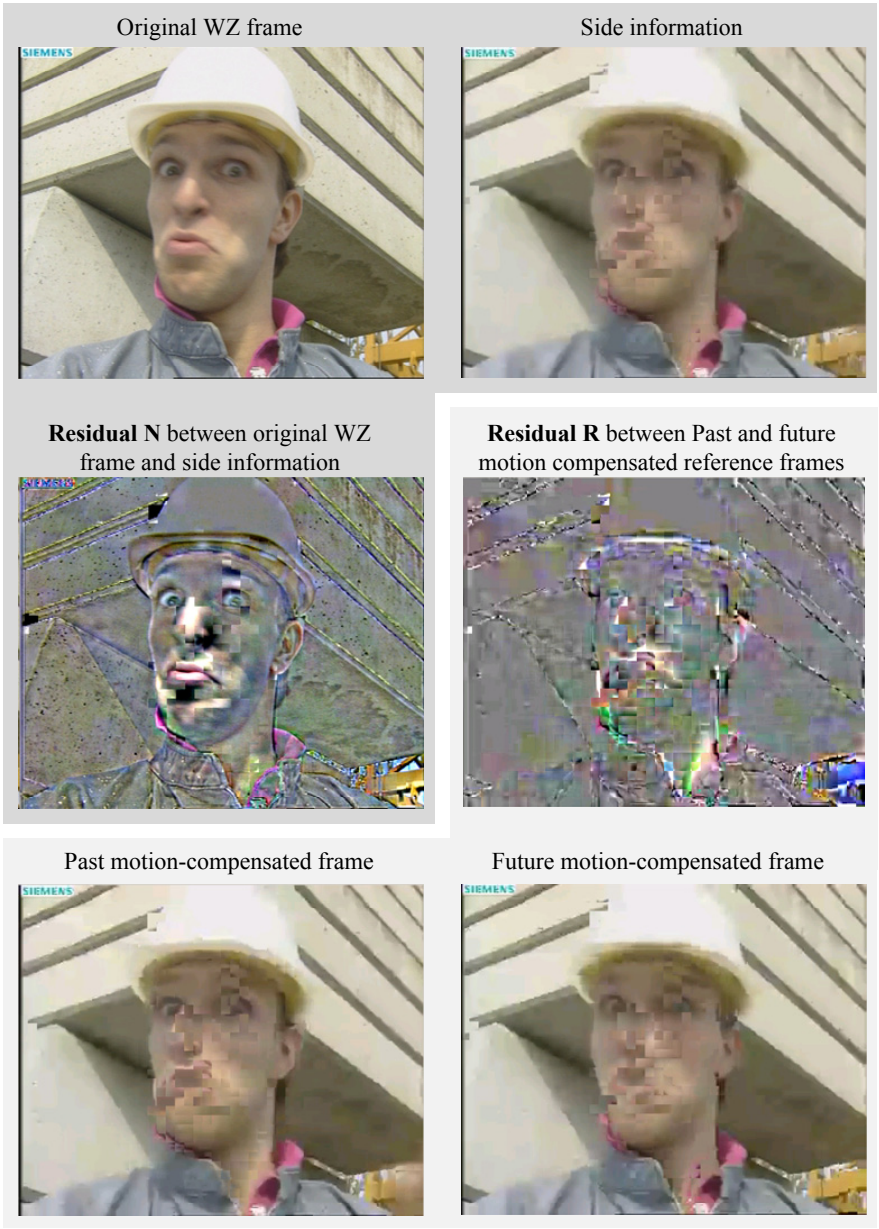

Fig. 5. Correlation noise N compared to the motion compensated residual R, for coarse quantization, i.e. an intra quantization parameter (IQP) of 40 is employed for the intra frames. In this case, some of the residual texture in N is not present in R, since quantization has removed this information from the reference frames.

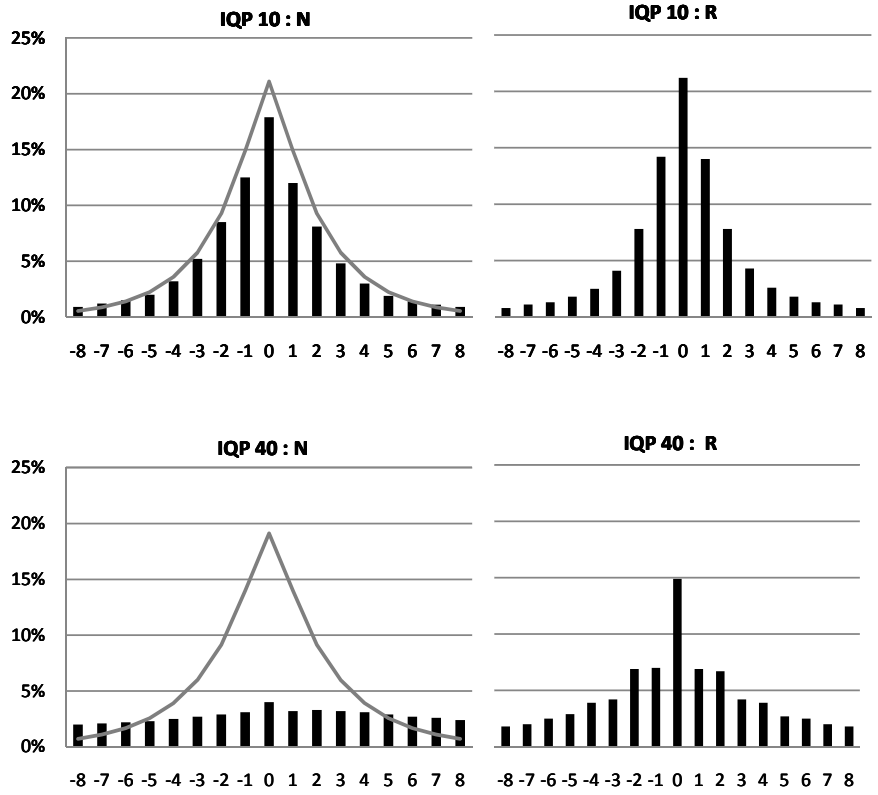

Fig. 6. Measured distribution of N and R, for the luma DC component of Foreman, for different quantization levels (H.264/AVC intra quantization parameter given). The average estimated distribution of N is drawn on top. Clearly, we can see that while the estimated noise is close to the true noise for fine quantization (IQP 10), there is a significant mismatch for coarse quantization (IQP 40).

since H.264/AVC intra decoded frames are generated anyway by the intra encoder in the context of mode decision and rate-distortion optimization.

To limit the overhead of sending the quantization noise variances to the decoder, some additional processing steps are performed. Firstly, it was observed that the variances of the chroma components (U and V) are usually very similar. Therefore, only the average of both is used. Next, the variances are quantized using a uniform quantizer having 2^M bins. It can be assumed that the variances are never much larger than the variance of a random variable that is uniformly distributed over the quantization bin, so that the quantizer range can be restricted to the interval $[0, d^2/12]$; d can be easily calculated from the H.264/AVC intra quantization parameter. For M, the largest integer is taken that is not greater than 5 and for which d is at least 1. Since a 4-by-4 DCT transformation is used, the result of processing the variances is that at most $5 \cdot (16 + 16) = 160$ bits need to be sent to the decoder per I frame.

Since the quantization noise statistics do not always change drastically from intra frame to intra frame, information is only sent if the average difference between the newly quantized

variances and the previously-sent quantized values is at least 0.5. This ensures that only significant updates are communicated.
In our experiments, the above processing steps proved to be efficient. The overhead of sending the quantization noise information to the decoder only accounts for maximum 0.05% of the total bit rate, for each rate point.

4.1.2 Decoder-side

At the decoder, the coded variances for the intra frames are received from the encoder, and reconstructed. Next, the decoder uses these variances to improve the modeling of the correlation noise between a particular Wyner-Ziv frame and the associated side information Y. As in most DVC systems, quantization of the intra and Wyner-Ziv frames is chosen in such a way that all frames have more or less the same quality. Therefore, it is assumed that the decoded intra frames and Wyner-Ziv frames are all affected by approximately the same noise. In addition, the quantization noise corrupting Y is similar to the noise in the reference frames, so that the quantization noise variances $\left(\sigma^Q_{(i,j)}\right)^2$ received from the encoder can be applied to the side information Y.
Now that an approximation of the quantization noise in Y has been obtained, this needs to be combined with the noise induced by motion, deformation, illumination changes, etc. Since the current methods for correlation noise estimation provide a good approximation when quantization noise is low (as shown before), both methods are combined. The standard deviation σ of the correlation noise N associated with a coefficient at index (i,j) in block k, is thus estimated as:

$$\sigma = \sigma^T_{k,(i,j)} + C \cdot \sigma^Q_{(i,j)}, \tag{11}$$

with

$$C = \begin{cases} 1 - \frac{\sigma^T_{(i,j)}}{2\sigma^Q_{(i,j)}} & \text{, if } \frac{\sigma^T_{(i,j)}}{2\sigma^Q_{(i,j)}} < 1 \\ 0 & \text{, otherwise} \end{cases} \tag{12}$$

where $\sigma^T_{k,(i,j)}$ and $\sigma^T_{(i,j)}$ relate to Equation 8 and Equation 10, respectively, since:

$$\sigma^T_{k,(i,j)} = \frac{\sqrt{2}}{\alpha^T_{k,(i,j)}}, \tag{13}$$

and

$$\sigma^T_{(i,j)} = \frac{\sqrt{2}}{\alpha^T_{(i,j)}}. \tag{14}$$

To justify this experimentally derived formula, similar measurements are performed as before. Comparing the new model for correlation noise estimation to the actual correlation noise in Figure 7 clearly shows that our estimation has become significantly more accurate.

4.2 Results

In order to quantify the impact of the proposed model on the coding performance, tests have been performed on sequences with different motion characteristics, including Mother and

		Proposed		Previous work		BJM delta rate	
		PSNR (dB)	rate (kbps)	PSNR (dB)	rate (kbps)	PSNR (dB)	rate (%)
MD	Q0	41.8	513	41.8	540	41	-6.1
	Q1	38.9	261	38.9	288	38	-11.6
	Q2	36.4	129	36.3	151	36	-16.6
	Q3	34.3	69	34.2	88	35	-19.5
Tab. Tennis	Q0	37.5	1666	37.5	1668	37	0.1
	Q1	33.4	827	33.3	835	33	-2.0
	Q2	30.2	405	30.1	421	30	-6.4
	Q3	27.8	216	27.7	235	28	-10.3
Foreman	Q0	38.2	1434	38.1	1445	38	-1.4
	Q1	34.8	708	34.7	722	34	-3.9
	Q2	31.6	357	31.4	374	31	-7.2
	Q3	28.7	188	28.6	201	29	-9.1

Table 2. Average results per Wyner-Ziv frame, for Mother and Daughter (MD), Table Tennis (Tab. Tennis), and Foreman. Bjøntegaard delta rate metrics (BJM) illustrate the evolution of the gain for different levels of quality (negative values indicate decrease in bit rate).

Daughter, Foreman, and Table Tennis. All are in CIF resolution, 30 fps, and with a GOP length of 4. The results are given in Table 2 for the Wyner-Ziv frames only.
The Bjøntegaard (2002) delta metric is used to illustrate the rate difference at a given level of quality. This metric shows that our new technique performs particularly well at low rates (i.e., coarse quantization), with average rate gains up to 19.5% per Wyner-Ziv frame for Mother and Daughter.
The results show that the gain for Mother and Daughter is larger than for Table Tennis and Foreman. This is because Mother and Daughter is a sequence with low motion characteristics, hence, the side information generation process is able to find very good matches, resulting in small values for R and consequently for $\sigma^T_{k,(i,j)}$ (and $\sigma^T_{(i,j)}$). Therefore, $\sigma^Q_{(i,j)}$ is relatively large, which results in a large impact on σ. For sequences with high motion content, $\sigma^T_{k,(i,j)}$ and $\sigma^T_{(i,j)}$ are larger so that the impact of our update is smaller.
The results obtained for our technique are interesting, but some areas still need further exploring. For example, we have assumed so far that the quantization noise in the decoded intra frames and the decoded Wyner-Ziv frames is similar. This might not always be true since the reconstruction of the intra frames and the reconstruction of the Wyner-Ziv frames is performed using different techniques.

4.3 Conclusions

The results of this section show that relying only on the motion-compensated residual between the reference frames (used for generating the side information) does not always deliver an accurate estimation of the correlation noise. Instead, we have shown that the quantization distortion in the reference frames needs to be taken into account as well in order to improve the accuracy of the noise estimates. Exploiting this information has resulted in significant performance gains, in particular at medium and low rates.
On the one hand, the results in this section underline the importance of accurate correlation noise modeling in DVC. On the other hand, the results encourage researchers to investigate

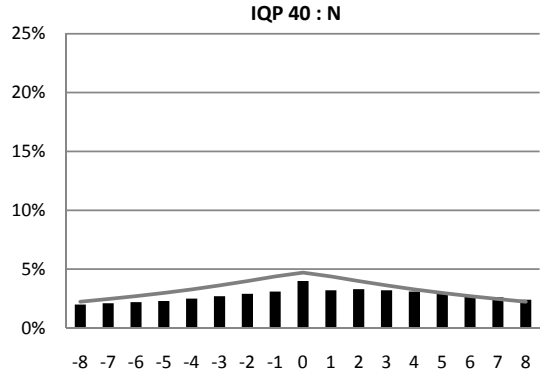

Fig. 7. The new method for correlation noise estimation is more accurate for coarse quantization (i.e. low rates).

other sources of information to improve the noise estimation accuracy, and develop new correlation noise models.

5. Compensating for motion estimation inaccuracies

This section details a second major contribution of this chapter, where the correlation model is further improved by compensating for the inaccuracies in the generation of the side information. This is achieved by using a correlation model based on multiple predictors, as detailed next.

Current techniques for side information generation commonly assume that the motion between the past and future reference frames can be approximated as linear. This assumption is made in, for example, Stanford's DVC architecture (Aaron et al. (2004a)) as well as in DISCOVER (Artigas et al. (2007)). Even in cases where more complex motion models are used, motion interpolation is still performed in a linear fashion. For example, Kubasov & Guillemot (2006) use mesh-based techniques to obtain a model of the motion between the past and future reference frame. The side information is then created through linear interpolation along the motion trajectories described by this model, assuming uniform motion between the reference frames.

The assumption that the motion between the reference frames can be approximated as linear becomes less valid when the distance between the reference frames increases. This is illustrated for a GOP of size eight. Side information has been generated for the 5th frame of the Foreman sequence, using the first frame as a past reference, and the 9th frame as a future reference. When analyzing the residual between the side information and the original frame in Figure 8, it is clear that a lot of errors need to be corrected, increasing significantly the Wyner-Ziv rate. Judging from the quality of the side information itself, it could already be expected that the accuracy of estimating the face is low. However, the residual also reveals that errors need to be corrected in the background as well. More specifically, we can see that edges in the side information are not predicted accurately. This is due to non-linear camera motion.

To compensate for inaccuracies in side information generation, most recent techniques apply a refinement approach. Decoding is performed partially, and the partially decoded frame is used to improve the quality of the side information. The improved side information is

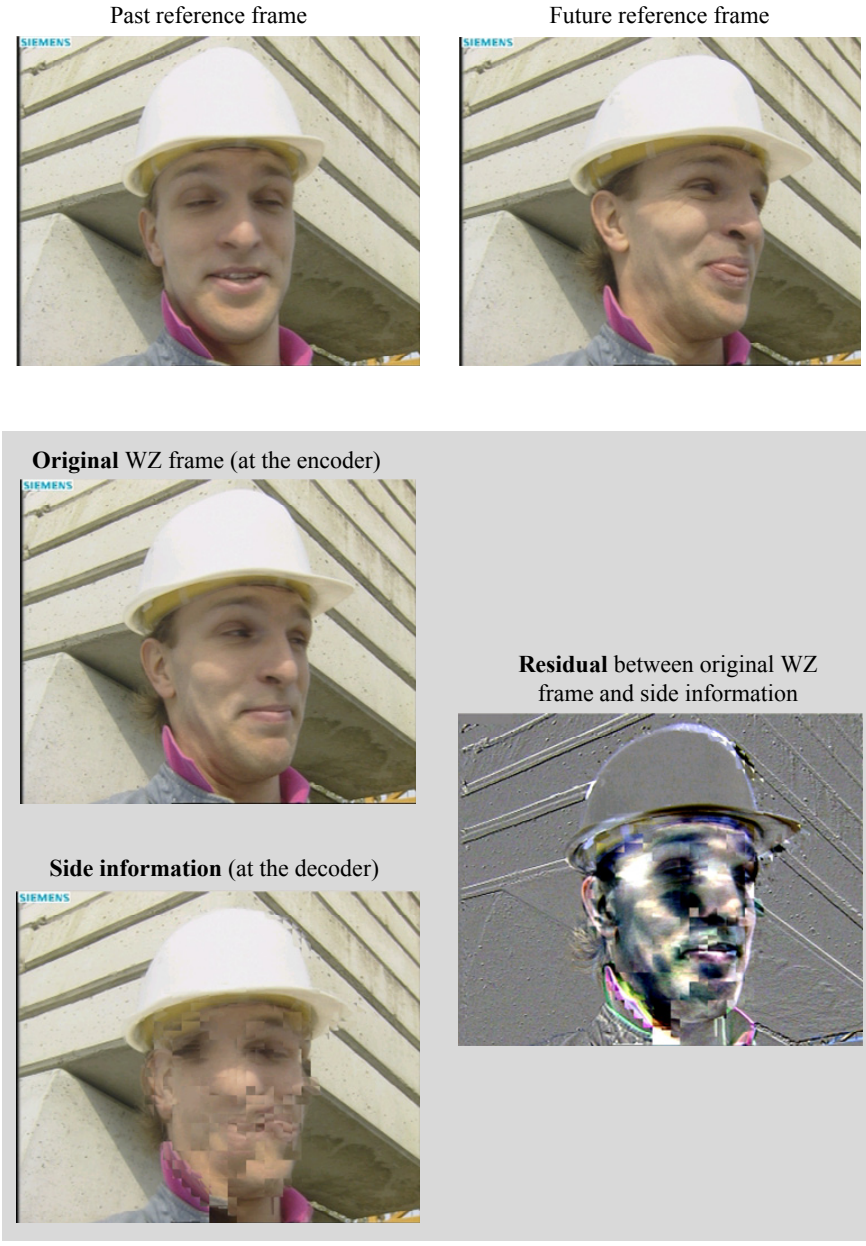

Fig. 8. Especially for large GOP's, the assumption of linear motion becomes less valid.

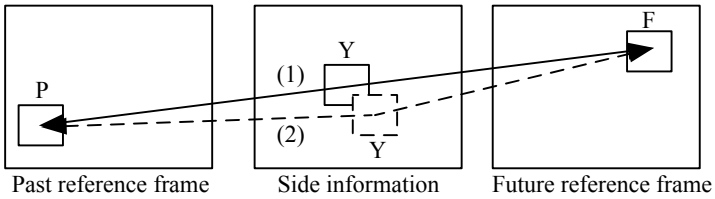

Fig. 9. The linear motion vector (1) could be inaccurate in the sense that the interpolation between P and F should be located on a different spatial position (2) than the one given by a linear motion vector (1).

then used for further decoding. For example, Martins et al. (2009) propose to refine the side information after each coefficient band has been decoded. A layered approach is used by Fan et al. (2009), dividing each frame into a low-resolution base layer and higher resolution refinement layers. The decoded base layer is used for improving side information accuracy of the following, higher resolution refinement layer, and so on. Information about the (partially) decoded frame can also be used to re-evaluate the side information generation process, for example, by identifying "suspicious" (i.e. possibly wrong) motion vectors (Ye et al. (2009)).
While these techniques show good results, what they have in common is that they can compensate for mistakes only *after* some information has been decoded. Therefore, in this section, a technique is proposed where some of these uncertainties are quantified *prior to decoding*. This is realized by extending the correlation model, improving the performance with an additional 8% gain in bit rate.

5.1 Proposed technique

The main idea is to compensate for inaccuracies by using more than one prediction for each block. We recall that a particular block in the side information Y is generated by averaging past and future reference blocks P and F respectively, using a linear motion vector. However, if the motion is non-linear, then the prediction should appear on a different spatial position in the side information (Figure 9). Hence, to predict a block at position (x_0, y_0), the block at position (x_0, y_0) in Y can be used, together with some of the surrounding blocks in Y. This strategy can also be beneficial in other cases with complex motion such as occlusion and deformation.
Before explaining this method, a description of the codec is provided.

5.1.1 Codec description

The proposed codec is depicted in Figure 10, highlighting the extensions that enable compensation for motion estimation inaccuracies.
As before, the techniques for side information generation are adopted from DISCOVER. The output of this process is the side information Y, and for each block, the (linear) motion vector MV_{SI}, as well as the residual R_{SI} between the past and future reference blocks. This information is used as input for the proposed extensions. First, for each block, multiple predictors are generated (Section 5.1.2). Next, each of these predictors is assigned a weight (Section 5.1.3), and the correlation between the predictors and the original is modeled (Section 5.1.4). This distribution is used by the turbo decoder, which requests bits until the decoded result is sufficiently reliable. Finally, the quantized coefficients are reconstructed (Section 5.1.5) and inverse transformed to obtain the decoded frame W'.

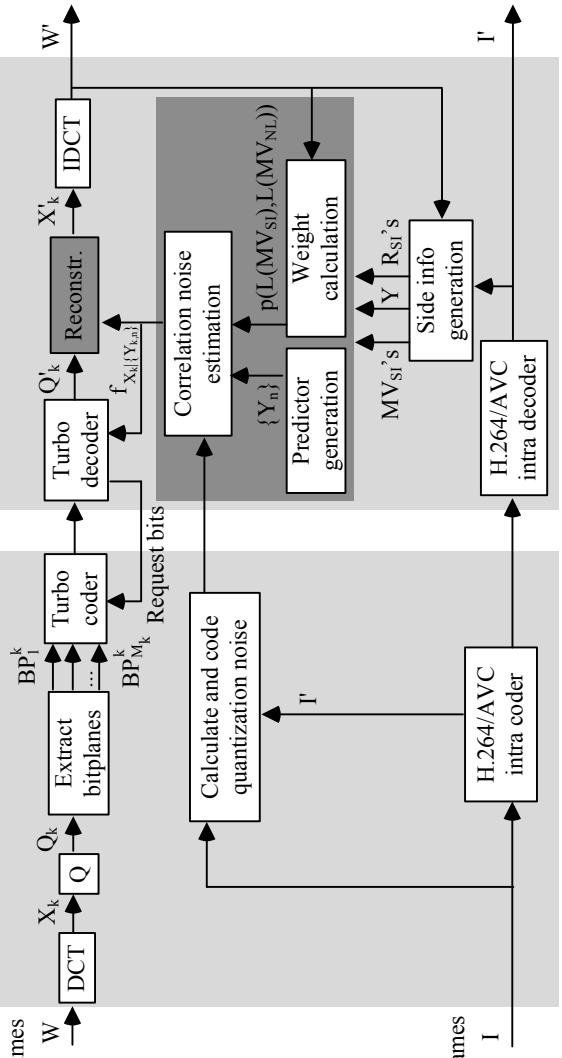

Fig. 10. The DVC codec featuring a correlation model that compensates for quantization noise and motion estimation inaccuracies.

5.1.2 Generation of predictors

A block at position (x_0, y_0) is predicted using multiple predictors, obtained from the side information frame Y. The first predictor is the predictor corresponding to linear motion, i.e., the block at the co-located position in Y. To compensate for motion inaccuracies such as non-linear motion, surrounding blocks in Y are taken into account as well. As a compromise between complexity and performance, eight additional predictors are used, namely the ones corresponding to positions $(x_0 \pm m, y_0 \pm m)$ in Y $(m \in \{0,1\})$. This results in a total of 9 predictors per block.

Not every predictor is equally likely, so that weights are calculated for each predictor, as explained in the following section.

5.1.3 Online calculation of the predictor weights

Each of the 9 predictors is assigned a weight, according to the probability that this predictor is the best one out of the set. This probability is estimated using the results from previously decoded frames. In a previously decoded frame W', given the previous side information Y, the best predictor for a block is obtained using the following procedure.

Each block in W' is compared to each of the 9 predictors in Y. More specifically, the mean absolute difference (MAD) is calculated between the block at a certain position (x_0, y_0) in W' and the co-located block in Y. This MAD indicates the amount of errors corrected when using the linear predictor. Likewise, the MAD for other predictors is calculated, for example, comparing the block at position (x_0, y_0) in W' to the block at position $(x_0 + 1, y_0 + 1)$ in Y etc. The predictor with the lowest MAD is then considered the best one out of the set.

However, a non-linear predictor is only considered best in case its MAD is lower than 0.9 times the MAD of the linear predictor. Otherwise, the linear predictor is considered to be the best. This criterion is used to ensure that only significant improvements over the linear predictor are taken into account. For example, in a region with not much texture, one of the non-linear predictors could have a lower MAD than the linear predictor, because the quantization noise in this predictor has distorted the block in such a way that it resembles better the decoded result. To avoid these situations, the MAD of a non-linear predictor must be lower than 0.9 times the MAD of the linear predictor. The value of 0.9 has been experimentally obtained.

Given the best predictor, a histogram table is updated, based on a characterization of the predictor using two parameters.

The first parameter is the amplitude of the motion. For example, the linear predictor could be highly reliable in static regions (e.g. in the background), but its reliability could be much lower for fast moving objects in the foreground. To discriminate between such cases, the amplitude of the (linear) motion vector MV_{SI} is used. To this extent, the following amplitude metric $L()$ is defined:

$$L((x,y)) = max(|x|,|y|). \tag{15}$$

The second parameter discriminates between different predictors, through the amplitude of the non-linearity of the predictor. Denote MV_{NL} as the predictor offset compared to the linear predictor. For example, if the linear predictor corresponds to the block at position (x_0, y_0) in Y, then the predictor at position $(x_0 + 1, y_0 - 1)$ in Y has $MV_{NL} = (1, -1)$.

Due to the use of the amplitude metric for this second parameter, all 8 non-linear predictors have a value of one for $L(MV_{NL})$. Only the linear-motion predictor has a different value, namely zero. As such, the statistics of the predictor having $MV_{NL} = (1, -1)$ are assumed to be similar to those of the predictor having $MV_{NL} = (0,1)$. This simplification can be refined

in future work, for example, by assigning higher weights to the non-linear predictors in the direction of the motion MV_{SI}.
Given the best predictor and its parameters $L(MV_{SI})$ and $L(MV_{NL})$, the histogram table T is updated. This table only covers the statistics of the current frame. All elements have been initialized to zero before any updating takes place. The parameters $L(MV_{SI})$ and $L(MV_{NL})$ serve as coordinates in T, and the corresponding value in T is incremented by one, for the best predictor.
After all blocks in W' have been processed, the result is combined with the result from previously decoded frames, by updating global statistics:

$$p_{i,j} = K \cdot p_{i,j} + (1 - K) \cdot \frac{T(i,j)}{\sum_k T(i,k)}, \tag{16}$$

where $p_{i,j}$ is a shorthand for $p(L(MV_{SI}) = i, L(MV_{NL}) = j)$. The update parameter K is set to 0.8. This value – which has been obtained through experiments – remains fixed for all test sequences. A more detailed study of the update parameter is left as future work.
The global statistics are used for calculating the weights for the predictors in the following Wyner-Ziv frame to be decoded. More specifically, the weight $w_{i,j}$ for a predictor characterized by $L(MV_{SI}) = i$, and $L(MV_{NL}) = j$ is calculated as:

$$w_{i,j} = \frac{p_{i,j}}{N_j}, \tag{17}$$

where N_j denotes the number of predictors (for that block) having a value of j for $L(MV_{NL})$. Hence, N_j equals one for the linear-motion predictor, and 8 for the remaining ones.

5.1.4 The correlation model

The goal is to model the correlation between the original X and the set of predictors denoted $\{Y_n\}$ (with $0 \leq n \leq 8$). This is modeled in the (DCT) transform-domain. For each 4-by-4 block in the original frame, 16 distributions are generated, i.e., one for each coefficient X_k (with $0 \leq k \leq 15$). The predictors are transformed, and all coefficients at the same index are grouped. Denote the predictors for X_k as $\{Y_{k,n}\}$.
As explained previously in this chapter, the correlation between the original and the side information is commonly modeled using a Laplace distribution. Hence, with multiple predictors, the conditional distribution $f_{X_k|\{Y_{k,n}\}}$ is modeled as a combination of weighted Laplace distributions, i.e.:

$$f_{X_k|\{Y_{k,n}\}}(x|\{y_{k,n}\}) = \sum_m w_m \cdot \frac{\alpha}{2} e^{-\alpha|x - y_{k,m}|}, \tag{18}$$

where $y_{k,m}$ indicates the k-th coefficient of the m-th predictor. w_m is the weight of the m-th predictor.
The scaling parameter α is calculated based on the reference residual of the linear predictor, using the techniques proposed in Section 4.

5.1.5 Coefficient reconstruction

After turbo decoding, the quantization bin q'_k containing the original value (with very high probability) is known at the decoder. The following step is to choose a value in this quantization bin as the decoded coefficient X'_k. This is done through optimal centroid reconstruction for multiple predictors (Kubasov et al. (2007)):

$$X'_k = \frac{\sum_m w_m \int_{q'_L}^{q'_H} x \cdot \frac{\alpha}{2} e^{-\alpha|x-y_{k,m}|} dx}{\sum_m w_m \int_{q'_L}^{q'_H} \frac{\alpha}{2} e^{-\alpha|x-y_{k,m}|} dx}, \tag{19}$$

where q'_L and q'_H indicate the low and high border of q'_k, respectively.

5.2 Results

Tests have been conducted on three different sequences: Foreman, Football and Coastguard. All are in CIF resolution, at a frame rate of 30 frames per second. A GOP of size 8 is used, and only the luma component is coded to enable a comparison with the DISCOVER codec. The system is also compared to our previous improvement described in Section 4, which uses only one predictor per block.

The results in Figure 11 indicate improvements over both systems. The gains are the largest for sequences with complex motion such as Football and Foreman, where the linear predictor does not always provide an accurate prediction. In these cases, using multiple predictors to compensate for inaccuracies shows average Bjøntegaard (2002) quality gains up to 0.4 dB over our approach in the previous section, and 1.0 dB over DISCOVER (both for Football and Foreman).

The gain diminishes for sequences with rather simple motion characteristics such as Coastguard. For such sequences, an accurate prediction is already provided by the linear-motion predictor, and little is gained by using additional predictors. Over our approach in the previous section, average quality gains of 0.1 dB are reported, and 1.4 dB over DISCOVER.

6. Conclusions

Modeling the correlation between the original frame available at the encoder and its prediction available at the decoder, is an important but difficult problem in DVC. While most current techniques rely on the motion-compensated residual between the reference frames, in this chapter, two techniques have been proposed that improve the modeling accuracy over the conventional approaches by using more than this residual.

The first technique exploits information about the quantization of the reference frames. In brief, information about the quantization of the key frames is sent from the encoder to the decoder. Using this information the correlation model at the decoder is refined, yielding high gains at medium and low rates. This method can be further improved, for example, by estimating the quantization noise at the decoder-side instead of sending this information from encoder to decoder. Another extension could be to account for spatial variation of the quantization noise.

The second technique presented in this chapter applies a correlation model with multiple predictors. In this solution we compensated for uncertainties in the assumption of linear motion between the reference frames by considering surrounding positions as well. Fair compression gains were reported especially for sequences featuring complex motion characteristics. We believe that this model can be further extended, for example, by explicitly dealing with occlusions.

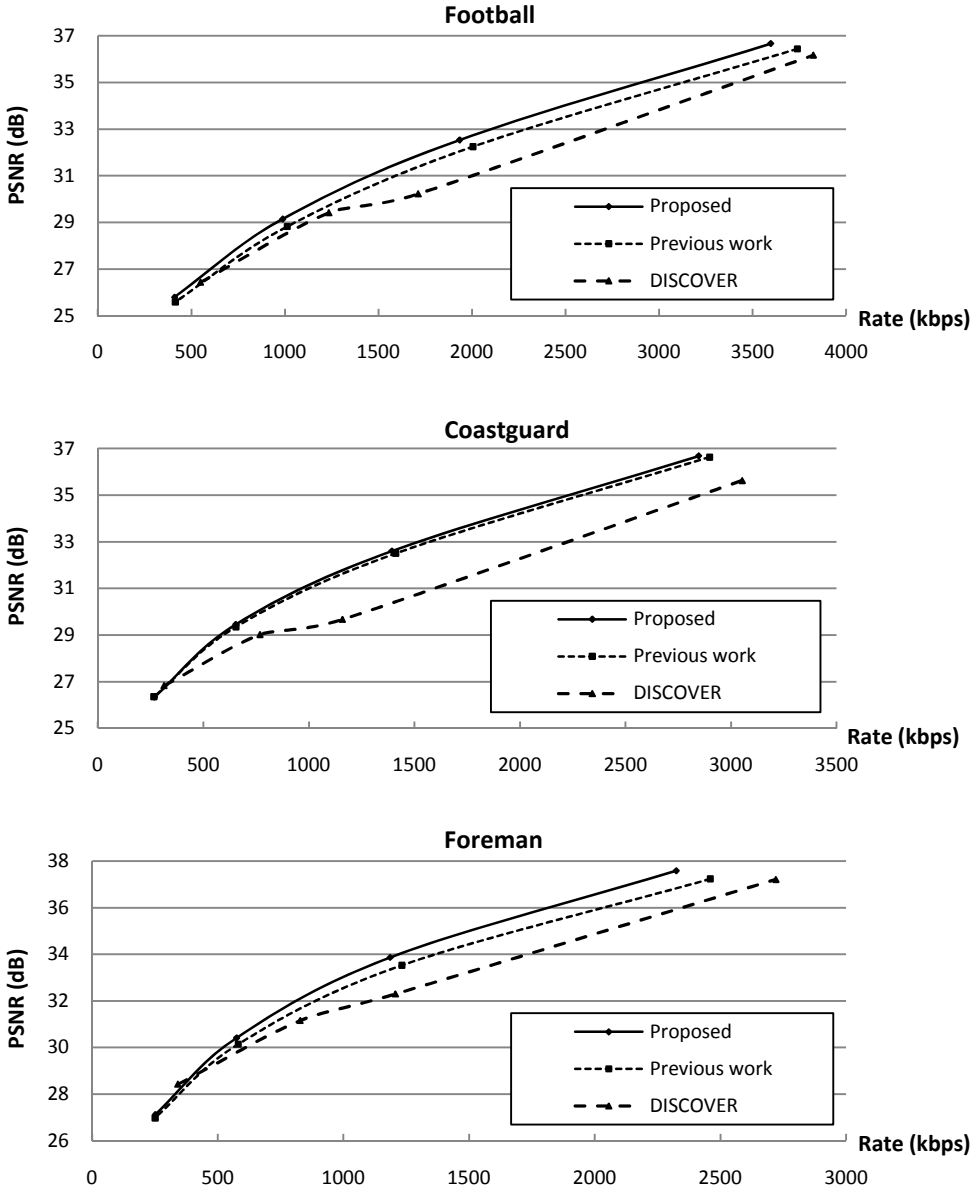

Fig. 11. Average quality gains of 0.4 dB are reported over our approach in the previous section, for both Football and Foreman. As expected, the gain diminishes for sequences with regular motion characteristics, such as Coastguard.

7. References

Aaron, A., Rane, S., Setton, E. & Girod, B. (2004). Transform-domain Wyner-Ziv codec for video, *Proc. SPIE Visual Communications and Image Processing*, Vol. 5308, pp. 520–528.

Aaron, A., Setton, E. & Girod, B. (2003). Towards practical Wyner-Ziv coding of video, *Proc. IEEE International Conference on Image Processing (ICIP)*, pp. 869–872.

Aaron, A., Zhang, R. & Girod, B. (2004). Wyner-Ziv video coding with hash-based motion compensation at the receiver, *Proc. IEEE International Conference on Image Processing (ICIP)*, pp. 3097–3100.

Artigas, X., Ascenso, J., Dalai, M., Klomp, S., Kubasov, D. & Ouaret, M. (2007). The DISCOVER codec: Architecture, techniques and evaluation, *Proc. Picture Coding Symposium (PCS)*.

Bjøntegaard, G. (2002). Calculation of average PSNR differences between RD-curves, *Technical report*, VCEG. Contribution VCEG-M33.

Brites, C. & Pereira, F. (2008). Correlation noise modeling for efficient pixel and transform domain Wyner-Ziv video coding, *IEEE Transactions on Circuits and Systems for Video Technology* 18: 1177–1190.

Fan, X., Au, O. C. & Cheung, N. M. (2009). Adaptive correlation estimation for general Wyner-Ziv video coding, *IEEE International Conference on Image Processing (ICIP)*.

Fan, X., Au, O. C., Cheung, N. M., Chen, Y. & Zhou, J. (2009). Successive refinement based Wyner-Ziv video compression, *Signal Processing: Image Communcation* . doi:10.1016/j.image.2009.09.004.

Gersho, A. & Gray, R. M. (1992). *Vector quantization and signal compression*, Kluwer Academic Publishers.

Girod, B., Aaron, A., Rane, S. & Rebollo-Monedero, D. (2005). Distributed Video Coding, *Proc. IEEE, Special Issue on Video Coding and Delivery*, Vol. 93, pp. 71–83.

Huang, X. & Forchhammer, S. (2009). Improved virtual channel noise model for transform domain Wyner-Ziv video coding, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 921–924.

Kubasov, D. & Guillemot, C. (2006). Mesh-based motion-compensated interpolation for side information extraction in Distributed Video Coding, *Proc. IEEE International Conference on Image Processing (ICIP)*.

Kubasov, D., Lajnef, K. & Guillemot, C. (2007). A hybrid encoder/decoder rate control for a Wyner-Ziv video codec with a feedback channel, *IEEE MultiMedia Signal Processing Workshop*, pp. 251–254.

Kubasov, D., Nayak, J. & Guillemot, C. (2007). Optimal reconstruction in Wyner-Ziv video coding with multiple side information, *IEEE MultiMedia Signal Processing Workshop*, pp. 183–186.

Lin, S. & Costello, D. J. (2004). *Error control coding*, 2 edn, Prentice Hall.

Macchiavello, B., Mukherjee, D. & Querioz, R. L. (2009). Iterative side-information generation in a mixed resolution wyner-ziv framework, *IEEE Transactions on Circuits and Systems for Video Technology* 19(10): 1409–1423.

Martins, R., Brites, C., Ascenso, J. & Pereira, F. (2009). Refining side information for improved transform domain wyner-ziv video coding, *IEEE Transactions on Circuits and Systems for Video Technology* 19(9): 1327–1341.

Trapanese, A., Tagliasacchi, M., Tubaro, S., Ascenso, J., Brites, C. & Pereira, F. (2005). Improved correlation noise statistics modeling in frame-based pixel domain Wyner-Ziv video coding, *International Workshop on Very Low Bitrate Video*.

Škorupa, J., Slowack, J., Mys, S., Lambert, P., Grecos, C. & Van de Walle, R. (2009). Stopping criterions for turbo coding in a Wyner-Ziv video codec, *Proc. Picture Coding Symposium (PCS)*.

Škorupa, J., Slowack, J., Mys, S., Lambert, P. & Van de Walle, R. (2008). Accurate correlation modeling for transform-domain Wyner-Ziv video coding, *Proc. Pacific-Rim Conference on Multimedia (PCM)*, pp. 1–10.

Ye, S., Ouaret, M., Dufaux, F. & Ebrahimi, T. (2009). Improved side information generation for distributed video coding by exploiting spatial and temporal correlations, *EURASIP Journal on Image and Video Processing* 2009. Article ID 683510.